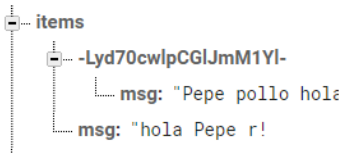
En Firebase existen dos tipos de listener:

1. ValueEventListener que puede ser asignado de dos formas con métodos del DatabaseReference:
    a. addValueEventListener, está siempre activo hasta que lo liberemos (consume recursos). Para liberar utilizaremos el método del DatabaseReference removeEventListener.
    b. addListenerForSingleValueEvent
2. ChildEventListener, cuando el nodo al que nos subscribimos tiene varios hijos. Permite notificar los siguientes eventos:onChildAdded, onChildChanged, onChildRemoved, onChildMoved y onChildCanceled

En este proyecto tendremos tres ejercicios incluidos. El primero carga la vista activity_main.xml, el segundo activity_mainv2 y el tercer activity_mainv3.

En el primero nos subscribimos a un nodo del tipo ítems:



Como vemos poedos subscribirnos al nodo principal ítems o al nodo msg :

```
public class MainActivity extends AppCompatActivity {
    EditText etTexto;
    TextView tvSalida;
    Button btEnviar;
    ChildEventListener childEventListener;
    FirebaseDatabase database;
    Adapter mAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mainv3);
        //activity_main:escribir en firebase y dos escuchadores distintos
        etTexto=(EditText)findViewById(R.id.etTexto);
        tvSalida=(TextView)findViewById(R.id.tVSalida);
        btEnviar=(Button)findViewById(R.id.btEnviar);
        database = FirebaseDatabase.getInstance();
        DatabaseReference dato=FirebaseDatabase.getInstance().getReference().child("items").child("msg");
```

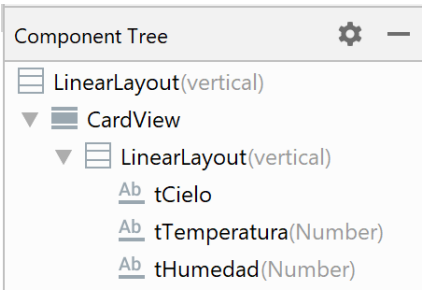
```
        btEnviar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (!etTexto.getText().toString().isEmpty())
                    database.getReference( path: "items").push().setValue(new Items(etTexto.getText().toString()));
            }
        });

        dato.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                //tvSalida.setText((String)dataSnapshot.getValue());
                Toast.makeText(getApplicationContext(),(String)dataSnapshot.getValue(),Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
        });
```

```
database.getReference( path: "items").addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
        tvSalida.setText(dataSnapshot.getValue().toString());
        //Toast.makeText(getApplicationContext(),dataSnapshot.getValue().toString(),Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
        //tvSalida.setText(dataSnapshot.getValue().toString());
        Toast.makeText(getApplicationContext(),dataSnapshot.getValue().toString(),Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {
        tvSalida.setText(dataSnapshot.getValue().toString());
    }

    @Override
    public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
        tvSalida.setText(dataSnapshot.getValue().toString()+" ha sido elimnado");
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {

    }
});
```

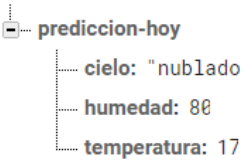
Este es otro ejemplo en el que cargamos un registro con tres campos (ejercicio2):

```
//activity_mainv2: escuchador registro
final EditText etCielo,etHumedad,etTemp;

etCielo=(EditText)findViewById(R.id.tCielo);
etHumedad=(EditText)findViewById(R.id.tHumedad);
etTemp=(EditText)findViewById(R.id.tTemperatura);
DatabaseReference dbPred=FirebaseDatabase.getInstance().getReference().child("prediccion-hoy");

dbPred.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        Prediccion p=dataSnapshot.getValue(Prediccion.class);
        etCielo.setText(p.getCielo());
        etTemp.setText(p.getTemperatura()+"°C");
        etHumedad.setText(p.getHumedad()+"%");
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {

    }
});
```

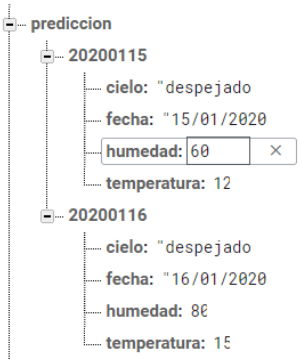
Y lo visualizamos en un layout con estos elementos:



En Firebase tendremos la siguiente estructura de almacenamiento Json:



La idea final es tener una lista con las predicciones de distintos días (ejercicio3):



Para ello tendremos un Holder del tipo:

```
public class PrediccionHolder extends RecyclerView.ViewHolder {
    EditText tCielo, tTemparatura,tHumedad;

    public PrediccionHolder(View v) {
        super(v);
        tCielo=(EditText) v.findViewById(R.id.tCielo);
        tHumedad=(EditText) v.findViewById(R.id.tHumedad);
        tTemparatura=(EditText) v.findViewById(R.id.tTemperatura);
    }

    public void bind(Prediccion item) {
        tCielo.setText(item.getCielo() );
        tTemparatura.setText(item.getTemperatura()+"°C");
        tHumedad.setText(item.getHumedad()+"%");
    }
}
```

Y un adaptador de la forma:

```
public class Adapter extends FirebaseRecyclerAdapter<Prediccion, PrediccionHolder> implements View.OnClickListener {
    private View.OnClickListener listener;

   Adapter(@NonNull FirebaseRecyclerOptions<Prediccion> options) { super(options); }

    @Override
    protected void onBindViewHolder(@NonNull PrediccionHolder holder, int position, @NonNull Prediccion model) {
       holder.bind(model);
    }

    @NonNull
    @Override
    public PrediccionHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
       View view= LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.card_view_re,viewGroup, attachToRoot: false);
       view.setOnClickListener(this);
       return new PrediccionHolder(view);
    }

    void onClickListener(View.OnClickListener listener) { this.listener=listener; }

    @Override
    public void onClick(View v) { if (listener!=null) listener.onClick(v); }
}
```

En nuestroMainActivity.java:

```
        //avtivity_mainv3:recyclerView
        final DatabaseReference dbPred=FirebaseDatabase.getInstance().getReference().child("prediccion");
        FirebaseRecyclerOptions<Prediccion> firebaseRecyclerOptions=new FirebaseRecyclerOptions.Builder<Prediccion>()
                .setQuery(dbPred,Prediccion.class).build();

        final RecyclerView recycler = (RecyclerView) findViewById(R.id.lstPredicciones);
        recycler.setHasFixedSize(true);
        recycler.setLayoutManager(new LinearLayoutManager( context: this));

        mAdapter = new Adapter(firebaseRecyclerOptions);
        recycler.setAdapter(mAdapter);
        //Click para eliminar elemento (se detecta en el holder)
        mAdapter.onClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(getApplicationContext(), text: "Elemento eliminado" +recycler.getChildAdapterPosition(view),
                        Toast.LENGTH_SHORT).show();
                String key=mAdapter.getRef(recycler.getChildAdapterPosition(view)).getKey();
                dbPred.child(key).removeValue();
            }
        });
    }

    @Override
    protected void onDestroy() {
        if(database!=null && childEventListener !=null)
            database.getReference( path: "items").removeEventListener(childEventListener);
        super.onDestroy();
    }

    @Override
    protected void onStart(){
        super.onStart();
        mAdapter.startListening();
    }

    @Override
    protected void onStop(){
        mAdapter.stopListening();
        super.onStop();
    }
```